

composer3d.py

```
1  from utils import *
2  from shape3d import Vertex, Shape2D, Shape3D
3
4  def Cube(_position : Vertex = Vertex(0, 0, 0), _edge : float = 1, _color : tuple[float] =
None):
5      """
6      Initialize 3D Cube object .
7      """
8      half_edge = _edge / 2
9      ...
10         3      2
11         0      1
12
13         7      6
14         4      5
15     ...
16     VERTICES = [
17         Vertex(
18             _position.x - half_edge,
19             _position.y - half_edge,
20             _position.z - half_edge,
21         ),
22         Vertex(
23             _position.x - half_edge,
24             _position.y + half_edge,
25             _position.z - half_edge,
26         ),
27         Vertex(
28             _position.x + half_edge,
29             _position.y + half_edge,
30             _position.z - half_edge,
31         ),
32         Vertex(
33             _position.x + half_edge,
34             _position.y - half_edge,
35             _position.z - half_edge,
36         ),
37
38         Vertex(
39             _position.x - half_edge,
40             _position.y - half_edge,
41             _position.z + half_edge,
42         ),
43         Vertex(
44             _position.x - half_edge,
45             _position.y + half_edge,
46             _position.z + half_edge,
47         ),
48         Vertex(
49             _position.x + half_edge,
50             _position.y + half_edge,
51             _position.z + half_edge,
52         ),
53         Vertex(
54             _position.x + half_edge,
55             _position.y - half_edge,
```

```

56         _position.z + half_edge,
57     ),
58
59 ]
60     return Shape3D(_position, VERTICES, [
61         Shape2D(_position, [
62             VERTICES[0],
63             VERTICES[1],
64             VERTICES[2],
65             VERTICES[3],
66         ], _color),
67         Shape2D(_position, [
68             VERTICES[4],
69             VERTICES[5],
70             VERTICES[6],
71             VERTICES[7],
72         ], _color),
73
74         Shape2D(_position, [
75             VERTICES[0],
76             VERTICES[1],
77             VERTICES[5],
78             VERTICES[4],
79         ], _color),
80         Shape2D(_position, [
81             VERTICES[2],
82             VERTICES[3],
83             VERTICES[7],
84             VERTICES[6],
85         ], _color),
86
87         Shape2D(_position, [
88             VERTICES[0],
89             VERTICES[3],
90             VERTICES[7],
91             VERTICES[4],
92         ], _color),
93         Shape2D(_position, [
94             VERTICES[1],
95             VERTICES[2],
96             VERTICES[6],
97             VERTICES[5],
98         ], _color),
99     ])
100
101
102
103 def Tetrahedron(_position : Vertex = Vertex(0, 0, 0), _edge : float = 1, _color :
tuple[float] = None):
104     """
105     Initialize 3D Tetrahedron object .
106     """
107     half_edge = _edge / 2
108     y_base = _edge / 4 * ((2 / 3) ** (1 / 2))
109     z_diff = _edge / (2 * (3 ** (1 / 2)))
110     '''
111         3
112
113         2
114     0      1

```

```

115     ...
116     VERTICES = [
117         Vertex(
118             _position.x - half_edge,
119             _position.y - y_base,
120             _position.z - z_diff,
121         ),
122         Vertex(
123             _position.x + half_edge,
124             _position.y - y_base,
125             _position.z - z_diff,
126         ),
127         Vertex(
128             _position.x,
129             _position.y - y_base,
130             _position.z + 2 * z_diff,
131         ),
132         Vertex(
133             _position.x,
134             _position.y + 3 * y_base,
135             _position.z,
136         ),
137     ]
138 ]
139 return Shape3D(_position, VERTICES, [
140     Shape2D(_position, [
141         VERTICES[0],
142         VERTICES[1],
143         VERTICES[2],
144     ], _color),
145     Shape2D(_position, [
146         VERTICES[0],
147         VERTICES[1],
148         VERTICES[3],
149     ], _color),
150     Shape2D(_position, [
151         VERTICES[1],
152         VERTICES[2],
153         VERTICES[3],
154     ], _color),
155     Shape2D(_position, [
156         VERTICES[2],
157         VERTICES[0],
158         VERTICES[3],
159     ], _color),
160 ])
161
162
163
164 def Octahedron(_position : Vertex = Vertex(0, 0, 0), _edge : float = 1, _color :
tuple[float] = None):
165     """
166     Initialize 3D Octahedron object .
167     """
168     half_edge = _edge / 2
169     height = _edge * ((1 / 2) ** (1 / 2))
170     ...
171         4
172
173         3             2

```

```

174         0         1
175
176         5
177     ...
178     VERTICES = [
179         Vertex(
180             _position.x - half_edge,
181             _position.y,
182             _position.z - half_edge,
183         ),
184         Vertex(
185             _position.x + half_edge,
186             _position.y,
187             _position.z - half_edge,
188         ),
189         Vertex(
190             _position.x + half_edge,
191             _position.y,
192             _position.z + half_edge,
193         ),
194         Vertex(
195             _position.x - half_edge,
196             _position.y,
197             _position.z + half_edge,
198         ),
199         Vertex(
200             _position.x,
201             _position.y + height,
202             _position.z,
203         ),
204         Vertex(
205             _position.x,
206             _position.y - height,
207             _position.z,
208         ),
209     ]
210
211     return Shape3D(_position, VERTICES, [
212         Shape2D(_position, [
213             VERTICES[0],
214             VERTICES[1],
215             VERTICES[4],
216         ], _color),
217         Shape2D(_position, [
218             VERTICES[0],
219             VERTICES[1],
220             VERTICES[5],
221         ], _color),
222
223         Shape2D(_position, [
224             VERTICES[1],
225             VERTICES[2],
226             VERTICES[4],
227         ], _color),
228         Shape2D(_position, [
229             VERTICES[1],
230             VERTICES[2],
231             VERTICES[5],
232         ], _color),
233     ]

```

```

234     Shape2D(_position, [
235         VERTICES[2],
236         VERTICES[3],
237         VERTICES[4],
238     ], _color),
239     Shape2D(_position, [
240         VERTICES[2],
241         VERTICES[3],
242         VERTICES[5],
243     ], _color),
244
245     Shape2D(_position, [
246         VERTICES[3],
247         VERTICES[0],
248         VERTICES[4],
249     ], _color),
250     Shape2D(_position, [
251         VERTICES[3],
252         VERTICES[0],
253         VERTICES[5],
254     ], _color),
255
256 ])
257
258
259 def Icosahedron(_position : Vertex = Vertex(0, 0, 0), _edge : float = 1, _color :
tuple[float] = None):
260     """
261     Initialize 3D Icosahedron object .
262     """
263     fi = pi * 2 / 5
264     r = _edge / (2 * sin(fi / 2))
265     y_top = _edge / 2 * (3 - 1 / sin(fi / 2)) ** (1 / 2)
266     circumradius = ((10 + (2 * 5 ** (1 / 2))) ** (1 / 2)) * (_edge / 4)
267     print(circumradius, y_top)
268     '''
269         5
270         3
271     4         2
272         0         1
273
274         6
275     7         10
276         8         9
277         11
278
279     '''
280     VERTICES = list()
281     for i in range(5):
282         vertex = Vertex(
283             _position.x,
284             _position.y + circumradius - y_top,
285             _position.z - r,
286         )
287         vertex.rotate(_position, fi * i, 'Ro_y')
288         VERTICES.append(vertex)
289     VERTICES.append(
290         Vertex(
291             _position.x,
292             _position.y + circumradius,

```

```

293         _position.z,
294     ))
295
296
297     for i in range(5):
298         vertex = Vertex(
299             _position.x,
300             _position.y - circumradius + y_top,
301             _position.z + r,
302         )
303         vertex.rotate(_position, fi * i, 'Ro_y')
304         VERTICES.append(vertex)
305     VERTICES.append(
306         Vertex(
307             _position.x,
308             _position.y - circumradius,
309             _position.z,
310         ))
311
312     return Shape3D(_position, VERTICES, [
313         Shape2D(_position, [
314             VERTICES[0],
315             VERTICES[1],
316             VERTICES[5],
317         ], _color),
318         Shape2D(_position, [
319             VERTICES[1],
320             VERTICES[2],
321             VERTICES[5],
322         ], _color),
323         Shape2D(_position, [
324             VERTICES[2],
325             VERTICES[3],
326             VERTICES[5],
327         ], _color),
328         Shape2D(_position, [
329             VERTICES[3],
330             VERTICES[4],
331             VERTICES[5],
332         ], _color),
333         Shape2D(_position, [
334             VERTICES[4],
335             VERTICES[0],
336             VERTICES[5],
337         ], _color),
338
339         Shape2D(_position, [
340             VERTICES[0],
341             VERTICES[1],
342             VERTICES[9],
343         ], _color),
344         Shape2D(_position, [
345             VERTICES[1],
346             VERTICES[2],
347             VERTICES[10],
348         ], _color),
349         Shape2D(_position, [
350             VERTICES[2],
351             VERTICES[3],
352             VERTICES[6],

```

```
353 ], _color),
354 Shape2D(_position, [
355     VERTICES[3],
356     VERTICES[4],
357     VERTICES[7],
358 ], _color),
359 Shape2D(_position, [
360     VERTICES[4],
361     VERTICES[0],
362     VERTICES[8],
363 ], _color),
364
365 Shape2D(_position, [
366     VERTICES[6],
367     VERTICES[7],
368     VERTICES[3],
369 ], _color),
370 Shape2D(_position, [
371     VERTICES[7],
372     VERTICES[8],
373     VERTICES[4],
374 ], _color),
375 Shape2D(_position, [
376     VERTICES[8],
377     VERTICES[9],
378     VERTICES[0],
379 ], _color),
380 Shape2D(_position, [
381     VERTICES[9],
382     VERTICES[10],
383     VERTICES[1],
384 ], _color),
385 Shape2D(_position, [
386     VERTICES[10],
387     VERTICES[6],
388     VERTICES[2],
389 ], _color),
390
391 Shape2D(_position, [
392     VERTICES[6],
393     VERTICES[7],
394     VERTICES[11],
395 ], _color),
396 Shape2D(_position, [
397     VERTICES[7],
398     VERTICES[8],
399     VERTICES[11],
400 ], _color),
401 Shape2D(_position, [
402     VERTICES[8],
403     VERTICES[9],
404     VERTICES[11],
405 ], _color),
406 Shape2D(_position, [
407     VERTICES[9],
408     VERTICES[10],
409     VERTICES[11],
410 ], _color),
411 Shape2D(_position, [
412     VERTICES[10],
```

```

413         VERTICES[6],
414         VERTICES[11],
415     ], _color),
416 ])
417
418
419 def Dodecahedron(_position : Vertex = Vertex(0, 0, 0), _edge : float = 1, _color :
tuple[float] = None):
420     """
421     Initialize 3D Dodecahedron object .
422     """
423     fi = pi * 2 / 5
424     r = _edge / (2 * sin(fi / 2))
425     circumradius = (3 ** (1 / 2)) * (1 + (5 ** (1 / 2))) * _edge / 4
426     _y_top = _edge / 2 * (3 * (3 + 2 * (5 ** (1 / 2))) / 2 - 1 / sin(fi / 2)) ** (1 / 2)
427
428     y_top = (circumradius ** 2 - r ** 2) ** (1 / 2)
429     d = _edge / 2 * (1 + (5 ** (1 / 2)))
430     r_ = d / (2 * sin(fi / 2))
431     y_mid = (circumradius ** 2 - r_ ** 2) ** (1 / 2)
432
433
434     # print(circumradius, y_top)
435     '''
436
437         3
438     4      2
439     0      1
440
441     8      7
442     9      6
443         5
444
445     '''
446     VERTICES = list()
447
448     # top
449     for i in range(5):
450         vertex = Vertex(
451             _position.x,
452             _position.y + y_top,
453             _position.z - r,
454         )
455         vertex.rotate(_position, fi * i, 'Ro_y')
456         VERTICES.append(vertex)
457
458     # top-mid
459     for i in range(5):
460         vertex = Vertex(
461             _position.x,
462             _position.y + y_mid,
463             _position.z - r_,
464         )
465         vertex.rotate(_position, fi * i, 'Ro_y')
466         VERTICES.append(vertex)
467
468     # bottom-mid
469     for i in range(5):
470         vertex = Vertex(

```



```

472         _position.x,
473         _position.y - y_mid,
474         _position.z + r_,
475     )
476     vertex.rotate(_position, fi * i, 'Ro_y')
477     VERTICES.append(vertex)
478
479 # bottom
480 for i in range(5):
481     vertex = Vertex(
482         _position.x,
483         _position.y - y_top,
484         _position.z + r,
485     )
486     vertex.rotate(_position, fi * i, 'Ro_y')
487     VERTICES.append(vertex)
488
489 return Shape3D(_position, VERTICES, [
490     Shape2D(_position, [
491         VERTICES[0],
492         VERTICES[1],
493         VERTICES[2],
494         VERTICES[3],
495         VERTICES[4],
496     ], _color),
497
498     Shape2D(_position, [
499         VERTICES[0],
500         VERTICES[1],
501         VERTICES[6],
502         VERTICES[-7],
503         VERTICES[5],
504     ], _color),
505     Shape2D(_position, [
506         VERTICES[1],
507         VERTICES[2],
508         VERTICES[7],
509         VERTICES[-6],
510         VERTICES[6],
511     ], _color),
512     Shape2D(_position, [
513         VERTICES[2],
514         VERTICES[3],
515         VERTICES[8],
516         VERTICES[-10],
517         VERTICES[7],
518     ], _color),
519     Shape2D(_position, [
520         VERTICES[3],
521         VERTICES[4],
522         VERTICES[9],
523         VERTICES[-9],
524         VERTICES[8],
525     ], _color),
526     Shape2D(_position, [
527         VERTICES[4],
528         VERTICES[0],
529         VERTICES[5],
530         VERTICES[-8],
531         VERTICES[9],

```

```

532     ], _color),
533
534     Shape2D(_position, [
535         VERTICES[-1],
536         VERTICES[-2],
537         VERTICES[-7],
538         VERTICES[6],
539         VERTICES[-6],
540     ], _color),
541     Shape2D(_position, [
542         VERTICES[-2],
543         VERTICES[-3],
544         VERTICES[-8],
545         VERTICES[5],
546         VERTICES[-7],
547     ], _color),
548     Shape2D(_position, [
549         VERTICES[-3],
550         VERTICES[-4],
551         VERTICES[-9],
552         VERTICES[9],
553         VERTICES[-8],
554     ], _color),
555     Shape2D(_position, [
556         VERTICES[-4],
557         VERTICES[-5],
558         VERTICES[-10],
559         VERTICES[8],
560         VERTICES[-9],
561     ], _color),
562     Shape2D(_position, [
563         VERTICES[-5],
564         VERTICES[-1],
565         VERTICES[-6],
566         VERTICES[7],
567         VERTICES[-10],
568     ], _color),
569
570     Shape2D(_position, [
571         VERTICES[-1],
572         VERTICES[-2],
573         VERTICES[-3],
574         VERTICES[-4],
575         VERTICES[-5],
576     ], _color),
577 ]))
578

```